DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION

DSTO

A New Data Acquisition System for the AMRL Low Speed Wind Tunnel

Owen Holland, Stephen Lam
and Yoel Link

DSTO-TR-0896

A New Data Acquisition System for the AMRL Low Speed Wind Tunnel

Owen Holland, Stephen Lam and Yoel Link

Air Operations Division
Aeronautical and Maritime Research Laboratory

DSTO-TR-0896

ABSTRACT

The data acquisition system in the Low Speed Wind Tunnel at the Aeronautical and Maritime Research Laboratory was recently upgraded. The MicroVAX II host computer was replaced by a Digital AlphaServer 400 running Digital UNIX, and the Bi-directional Parallel Interface data bus was replaced by ethernet and fast serial communication. The upgrade provides a system which is easier to use; includes a graphical user interface; provides a communication bus based on standard communication protocols, which achieves higher data transfer rates; is far more reliable and easy to maintain; and the system is more flexible than previous versions.

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Published by

DSTO Aeronautical and Maritime Research Laboratory
PO Box 4331
Melbourne Victoria 3001 Australia

Telephone: (03) 9626 7000
Fax: (03) 9626 7999

AR-011-129
November 1999

APPROVED FOR PUBLIC RELEASE

A New Data Acquisition System for the AMRL Low Speed Wind Tunnel

Executive Summary

Wind tunnels are one of the primary sources of aerodynamic data for aeronautical research. The Australian Defence Science and Technology Organisation (DSTO) operates two major wind tunnels at the Aeronautical and Maritime Research Laboratory (AMRL), one covers the low speed regime and the other covers the transonic speed regime. Results obtained from wind tunnel tests are used in many areas, such as aerodynamics research, aircraft design and modification, validation of computational fluid dynamics codes, and flight dynamic models.

Achieving a high level of accuracy in wind tunnel test results is essential. Accuracy of the results depends on many factors, such as the data acquisition system, the force and moment measurement system, the pressure measurement system, and an array of other sensors that may be used in gathering data.

The data acquisition system in the Low Speed Wind Tunnel (LSWT) at AMRL was recently upgraded. The MicroVAX II host computer was replaced by a Digital AlphaServer 400 running Digital UNIX, and the Bi-directional Parallel Interface data bus was replaced by ethernet and fast serial communication. The upgrade provides a system which is easier for staff to use; includes a graphical user interface; provides a communication bus based on standard communication protocols, which achieves higher data transfer rates and is far more reliable and easy to maintain; and the system is more flexible than previous versions.

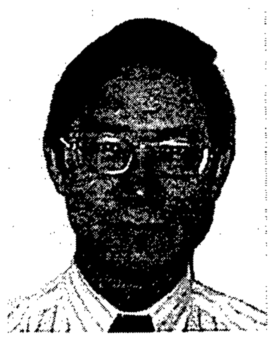
The LSWT data acquisition system is under constant development as software and hardware technology improves. Instrumentation modules, both VME-based and PC-based, have been developed and used to provide the functionality required for wind tunnel testing. Currently, a system built on the VXIbus standard is being investigated as a potential replacement for VME-based instrumentation modules.

The upgrade of the data acquisition system achieved all the objectives stated at the outset of the project. Most importantly, it provides DSTO and Defence customers with the ability to continue to obtain aerodynamic data in the low speed regime accurately and efficiently.

Authors

Owen Holland
Air Operations Division

Owen Holland graduated from Bendigo College of Advanced Education in 1977 with a Diploma of Electronic Engineering and then obtained a Bachelor of Engineering (Electronic) from Royal Melbourne Institute of Technology in 1983. In 1996 he was awarded a Master of Engineering in Electrical and Electronic Engineering from Victoria University of Technology. He commenced employment at the Aeronautical Research Laboratory in 1978. Since then he has worked in a number of areas including instrumentation and data acquisition systems for wind tunnels. He is currently a Senior Professional Officer in the Air Operations Division at the Aeronautical and Maritime Research Laboratory.



Stephen Lam
Air Operations Division

Dr Stephen Lam graduated as a Bachelor of Engineer (Mechanical) in 1979 and obtained a degree of Master of Engineering Science in 1982 from the University of Melbourne. He later undertook a research study on Natural Convection in Trapezoidal Cavities at Monash University and was awarded the degree of Doctor of Philosophy in 1990. Dr Lam joined DSTO in 1988 and has since been working in the area of wind tunnel research. He has implemented a machine to calibrate wind tunnel strain gauge balances, and took a leading role in the implementation of a new data acquisition system for both the Low Speed and Transonic Wind Tunnels at AMRL. Dr Lam is now actively involved in the implementation of advanced aerodynamic testing techniques, and is investigating the application of Pressure Sensitive Paint measurement technique in both the transonic and low speed flow regimes.

Yoel Link

Air Operations Division

Yoel Link completed his Bachelor of Science in 1987 and his Bachelor of Engineering in Aeronautical Engineering in 1989, both at Sydney University, and he joined the Aeronautical Research Laboratory at Melbourne the following year. He completed a Master of Business Administration in Technology Management in 1995 at Monash University. He has predominantly worked in Flight Mechanics and experimental aerodynamics in the Wind Tunnels. During this period he has accumulated experience in aerodynamics with the Jindivik, Tonic, PC-9, Mk82 store, Amphibious Transport (LPA) ship, and the Hydrographic Ship wind tunnel test programmes. He has also been responsible for the development of the wind tunnel data acquisition systems, and he has been involved with the Transonic Wind Tunnel Upgrade project.

Contents

1. Introduction

The most recent version, prior to the current upgrade, of the data acquisition system in the Low Speed Wind Tunnel (LSWT) at the Aeronautical and Maritime Research Laboratory (AMRL) was installed during the late 1980s (Matheson et al, 1991). The system was based on a host-slave concept, where the host was a Digital MicroVAX II computer, and the slaves were VME-based or PC-based instrumentation modules designed in-house at AMRL. The host computer issued commands to the modules, gathered all the data and performed the data reduction. Each module acquired data related to specific functions only, as required by the wind tunnel test programme. An AMRL-designed data bus provided bi-directional communication between the host computer and the slave modules (Harvey, 1989).

The Digital MicroVAX II operating system was VMS and the data processing software source code, written in Fortran, was based on code that was originally written for a Digital PDP 11/44. Over the years the source code underwent many modifications and had become difficult to maintain. A Digital AlphaServer 400 was chosen as the replacement for the host computer, and it provided a C, X/Motif, and OpenGL software development environment. Using this mixture of development tools the key design criteria for the system included platform independence and ease of maintenance.

The original bi-directional parallel interface data bus did not achieve its stated speed or reliability design objectives and it would suddenly stop working for unacceptable periods of time. In choosing a replacement for the data bus, minimum modification to the slave modules was added to reliability and speed as essential requirements. The communication medium chosen for the new data bus was the widely used 10 Megabit per second ethernet standard. All PC-based instrumentation modules were quickly and easily connected to the new data bus by replacing the bi-directional parallel interface card with an ethernet card. VME-based instrumentation modules were connected to a multi-port RS-232 serial card in a PC which translated the RS-232 data onto the ethernet data bus (Spataro and Kent, 1998).

This report provides a general overview of the components that make up the wind tunnel data acquisition system, including descriptions of both hardware and software. The design concepts and implementation described are applicable to many wind tunnel data acquisition systems.

2. System Overview

The overall wind tunnel data acquisition system concept is illustrated in Figure 1. The Digital AlphaServer 400 4/233 acts as the host computer, transmitting read and write requests to individual modules, and listening to data being broadcast on the dedicated

data acquisition ethernet network. The host is connected via a 16-port ethernet hub to the PC-based modules, which are connected directly to the ethernet using standard ethernet network cards. A multi-port serial hub PC acts as the gateway to the VME-based modules as shown in Figure 1 (Spataro and Kent, 1998). The serial hub PC interprets the ethernet requests from the host and converts them to serial requests for the specific VME-based module. Each instrumentation module performs a particular function as part of the overall data acquisition system and is connected to its own sensors, motors, controllers, and instrumentation using a range of hardware.

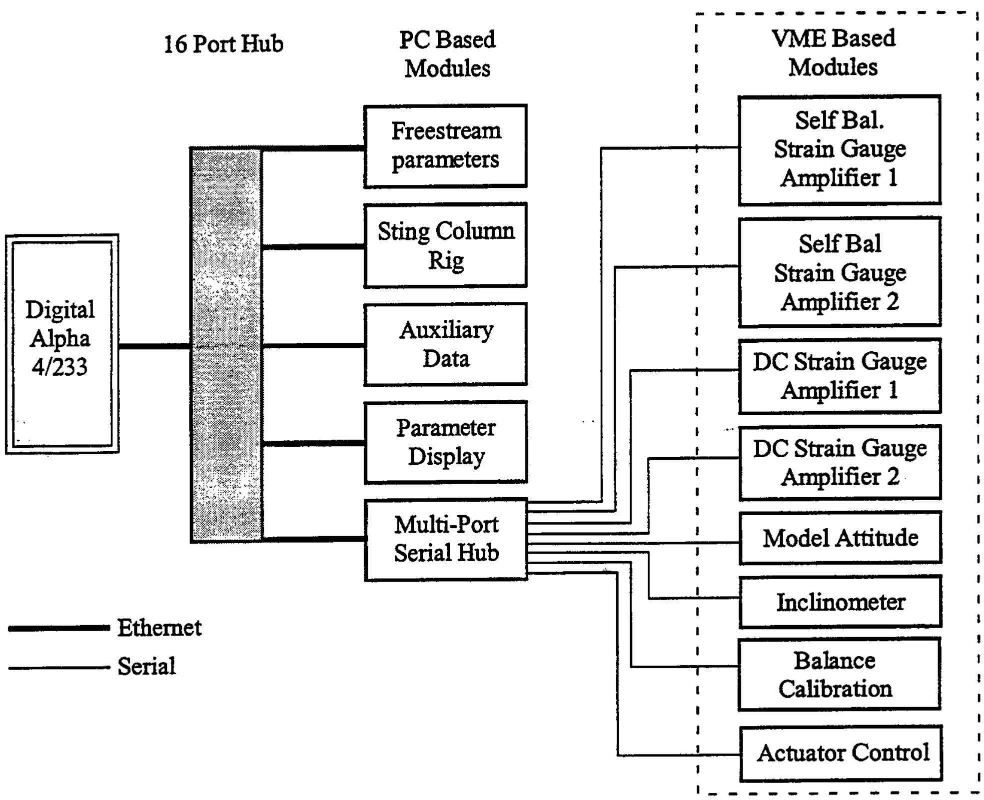


Figure 1 Data acquisition and instrumentation conceptual layout

3. Host Computer - Digital AlphaServer 400 4/233

The Digital Alpha Server 400 is an Alpha microprocessor 233 MHz CPU system. The server supports up to 384 MB of industry-standard SIMM memory and has an integrated Fast Narrow SCSI-2 controller. The system enclosure supports five storage devices including a floppy diskette drive, CD-ROM, and hard drives. Six industry

standard I/O expansion slots (two PCI, three ISA, and one PCI/ISA slot) provide for options such as high-performance graphics, networking and SCSI adapters.

The system configuration used in the LSWT data acquisition system includes 128 MB of memory, two 2.1 GB hard disk drives, a CD-ROM drive, an 8 GB DAT drive, two 10/100 ethernet controllers, and a ZLXp-E3 24-bit z-buffering graphics adapter. The operating system used is Digital UNIX V4.0E. The data acquisition software is written using the Digital C development tools, the graphical user interfaces are written using X/Motif, and the graphical displays providing real time monitoring of the data are written using a mixture of Open GL and GLUT (GL Utilities Toolkit) provided by the Digital Open3D software libraries.

4. Software Description

The data acquisition software consists of a number of different programs, all of which are accessible to the user from a single data acquisition window (Figure 2). Configuration of the test programme is required prior to starting a test and all required parameters must be input using the *wtsetup* software (Edwards, 1999). Real time graphical monitoring of test data is provided by the *monfrc* software (Edwards, 1999), and the graphical output is displayed on the main console. Data acquisition, reduction and storage functions are provided by the *comfrc* software (Edwards and Link, 1999), and is accessed via a PC running an X session (using the X-emulation software X-Win32[1]) from the host computer. The software used to acquire the data from the ethernet, *testBcast*, is described later in Section 7.

[1] Starnet Communications Corporation, V4.01, 1997. Further information is available at http://www.starnet.com/ on the World Wide Web.

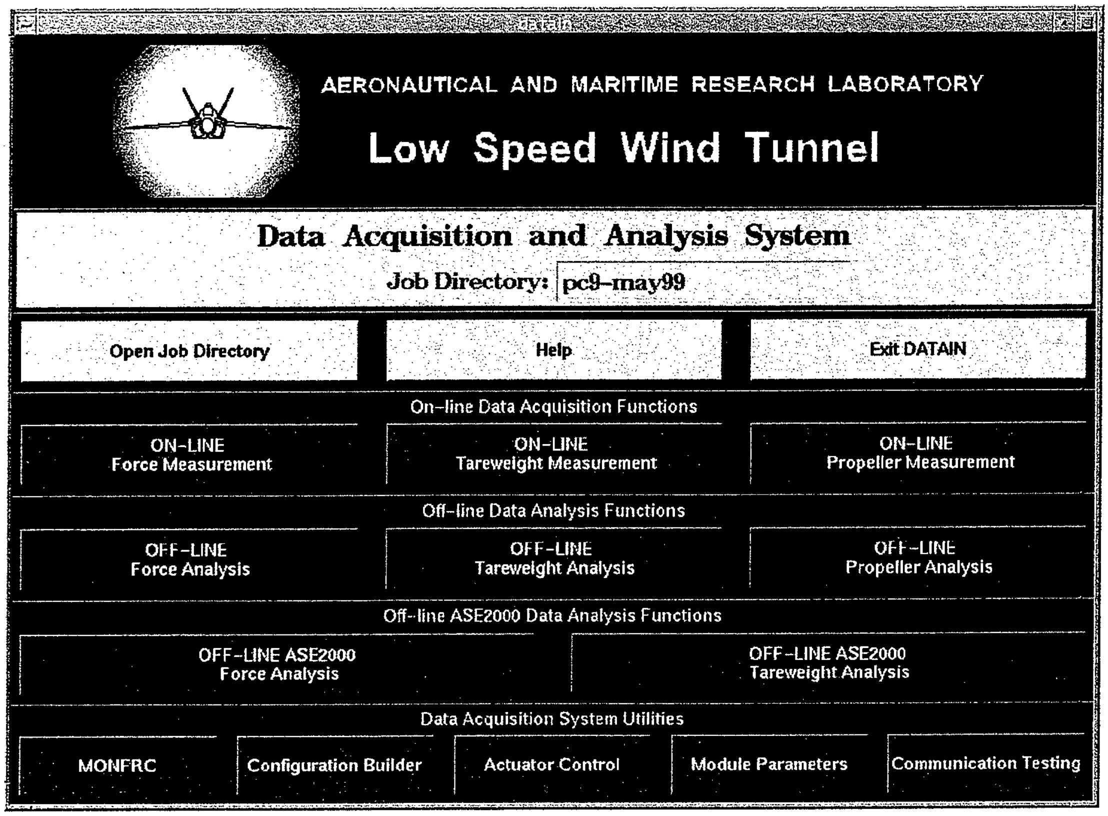


Figure 2 Main data acquisition interface window

4.1 Test Setup

The software *wtsetup* provides the user interface to the files required to configure a test programme. Parameters such as the chord, span, surface area, wind tunnel correction variables, balance calibration matrices, sting deflection matrices and many others, must be determined and input prior to commencing a wind tunnel test. Additionally, the user must determine and input the safe operating range for the wind tunnel test parameters, such as tunnel velocity, tunnel temperature, balance loads, and set these limits for the real time displays.

4.2 User Interface

The software *comfrc* (Figure 3) provides the user interface to the data acquisition process for force and moment wind tunnel tests. The user creates new data files or opens existing files for appending data to, 'takes data', and creates the output files containing data in engineering units, using this software. The configuration files created using *wtsetup* are read by *comfrc* and used in the data reduction functions.

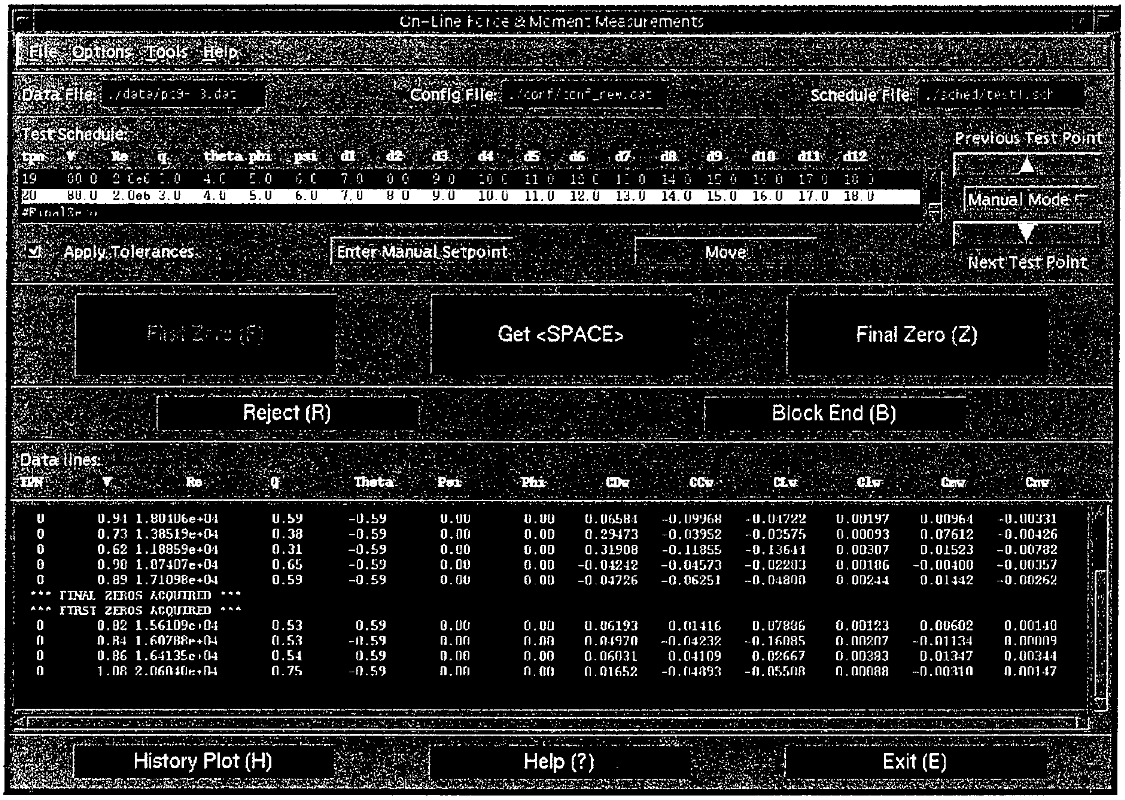


Figure 3 Comfrc main window – for force and moment tests

4.3 Real Time Data Monitoring

The software *monfrc* (Figure 4) provides the real time monitoring of wind tunnel operating and test parameters, including tunnel velocity, tunnel pressures and temperatures, test Reynolds number, model attitude, control surface deflection angles, and balance voltages and loads. The user, running the *comfrc* program, decides when the required data is 'on setpoint' using the monitoring software and initiates the data sampling and recording process. This procedure is semi-automated through the use of a schedule file which includes the test conditions for each data point. However, at present, the velocity in the test section is not automatically controlled; it is controlled by a single human-in-the-loop operator.

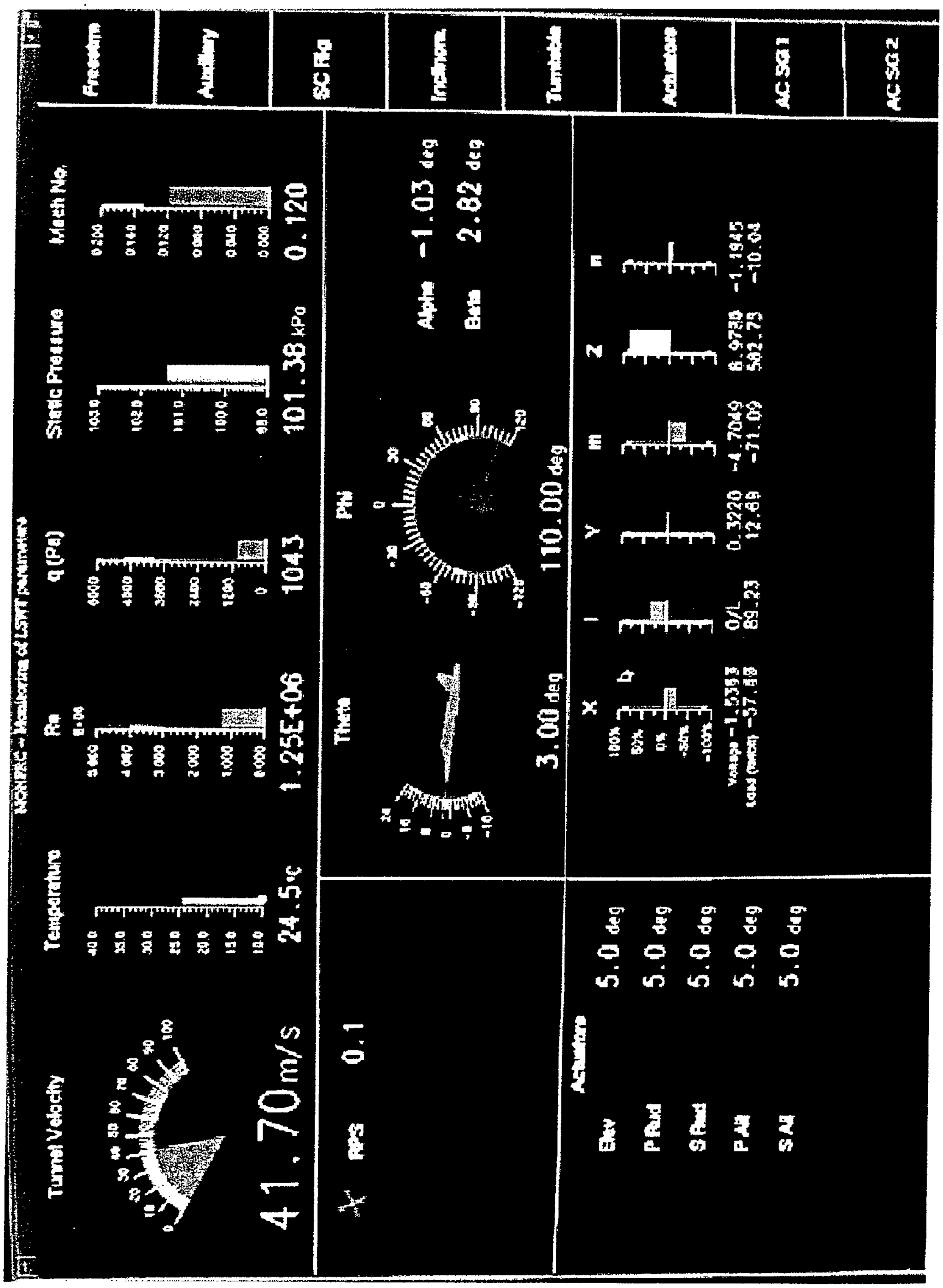


Figure 4 Monfrc main window – for real time monitoring of wind tunnel parameters

5. VME-based Instrumentation Modules

Each data acquisition function is primarily performed by an instrumentation module which is built around a microprocessor with local processing power. These VME-based modules use Motorola MC68000 microprocessors running proprietary assembly language software, with a VMEbus backplane bus system. In addition to the cards required for the specific data acquisition functions, the original design of the modules included a card providing the ability to communicate with the host computer via the bi-directional parallel interface (BPI) communication protocol.

A fundamental requirement of the data acquisition upgrade was to minimise the modifications to the VME-based modules' hardware and software. After the decision was taken to replace the BPI data bus with ethernet, an ethernet card compatible with the VME implementation used in the modules was sought. An ethernet card which potentially met the requirement was sourced and work commenced on attempting to integrate it with the VME-based module. The integration of the VME ethernet card was not successful for two main reasons. Firstly, incompatibility between the AMRL-designed VME-based modules, which were designed in the early 1980s and partially complied with the 'VME pre-Rev C.1 1985 version' (Motorola, 1985), and the ethernet card's VME implementation which was post-1987. Secondly, it appeared that the software integration of the ethernet hardware with the assembler code written for the VME-based modules would be a difficult process, and it would require additional real time operating systems and major re-writes of code. After three months of unsuccessful attempts, a decision was made to discontinue the work with the VME ethernet card.

A bus replacement option evaluated earlier in the project for the VME-based modules was the multi-port serial connection option. As a result of the unsuccessful attempt to integrate the VME ethernet card with the instrumentation modules, the multi-port serial connection option was chosen as the preferred method and was implemented. The BPI communication card in each module was replaced with an AMRL-designed serial communication card incorporating a Motorola MC68681 dual asynchronous receiver/transmitter (DUART) chip. The modules were connected to a PC containing a Hostess550 eight port RS-232 serial adapter card. This PC, known as the Serial Hub PC, also included an ethernet card and it was capable of communicating with the host computer over the dedicated data acquisition system ethernet network. The software residing on the VME-based modules was modified to provide the ability to transmit and receive serial data packets. (Spataro and Kent, 1998)

The communication methodology is predominantly based on data streaming, reading from vectors, writing to vectors, and a database which contains the most recent data from the module, as illustrated in Figure 5.

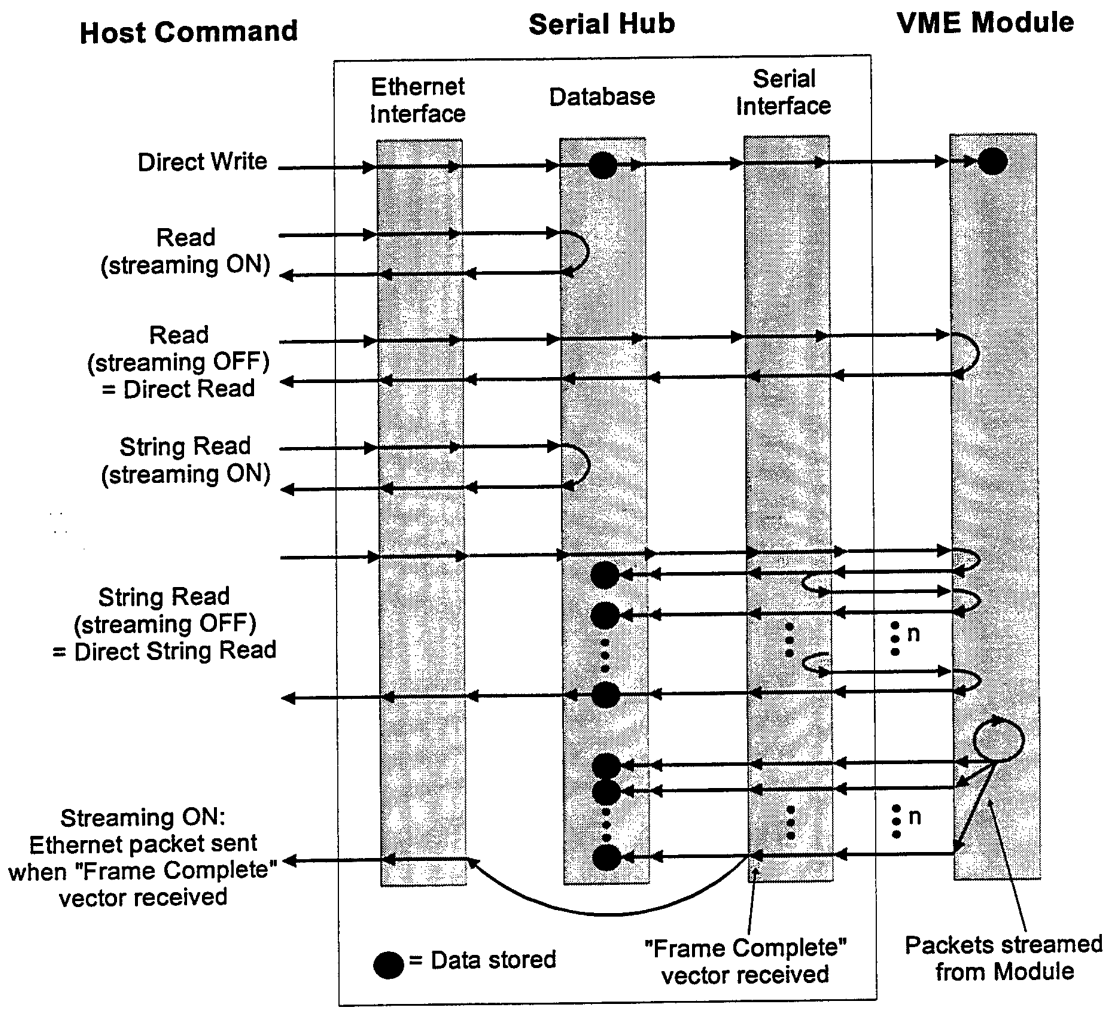


Figure 5 Command and data flow in the Serial Hub

In normal operating mode, at completion of the VME module's data trigger cycle, the module streams its current data to the serial hub PC, and the serial hub PC then updates its database. When the 'frame complete' vector is received from the module, indicating that a complete set of module data is now in the serial hub database, an ethernet packet is compiled and sent to the host computer. Single reads from the serial hub database, or direct reads from the module, are both valid and are handled by the serial hub. In addition, direct writes are supported. This allows initialisation data to be written to the module, as well as providing the ability to initiate control or command sequences which are pre-programmed in the VME modules.

An example of a VME-based module is the Model Attitude Module which controls the turntables located in the floor and ceiling of each of the two test sections. The

turntables are driven by electric motors and they can be rotated to enable a wind tunnel model to be yawed through ±185°. In addition to the CPU, memory and RS-232 serial interface cards, the module contains a resolver/encoder interface card, three parallel digital input/output cards and a relay drive card for computer control of the turntables. The module receives commands from the host computer via the serial hub PC to set and move to new yaw angles, and the current yaw angle value in the serial hub PC database is continually updated based on the signal output from the resolver. (Kent, 1998)

6. PC-based Instrumentation Modules

In the early 1990s, as PC technology became dominant, it was decided that the design of the instrumentation modules would utilise PCs in preference to the VME-based design. A BPI card was developed for the PC/AT bus architecture and four modules were designed and manufactured. PC-based modules were easier to maintain, modify, and expand as the wind tunnel instrumentation requirements changed and increased. The hardware change to accommodate the new data bus was a simple replacement of the BPI card with a standard PC ethernet card.

The operating system running on the PCs is MS-DOS V6.2. The application software is written in C or C++ and utilises RTKernel[2] (RTK), a real time multi-tasking software system. The software was modified to replace the BPI driver with code to communicate over ethernet with TCP/IP protocols.

The communication methodology is similar to that described in Section 4, except there is no need for an intermediate stage (serial hub PC) between the module and the host. In normal operating mode, the PC gathers data from sensors and other sources. On completion of a pre-defined stage of the data acquisition cycle an ethernet packet is assembled and broadcast or narrowcast (sent to a single IP address) over the ethernet. The host collects the packet and makes the data available to the *comfrc* and *monfrc* programs by placing the data in a shared memory database. Direct reads and direct writes are also available to provide full communication with each PC module.

An example of a PC-based module is the Freestream Parameter Module (Bird, 1999). This module monitors tunnel pressures and temperature, and calculates parameters such as tunnel velocity, dynamic pressure, Mach number, air density, and test Reynolds number. All parameters are broadcast over the network for other modules to use if required, but more specifically for the host computer running the monitoring software *monfrc* to update the display, as described in Section 4.3.

[2] RTKernel is a 16 bit Real Time Kernel supplied by On Time Infomatik GmbH. Further information is available at http://www.on-time.com/ on the World Wide Web.

7. Data Bus Description

The host computer is connected to the instrumentation modules via a dedicated 10BaseT[3] ethernet network that replaced the bi-directional parallel interface (BPI) data bus (Harvey, 1989). With the BPI data bus the host controlled all data transfer, and data was transferred by repeated cycles of the host (master) reading or writing sixteen bit words to individual slave modules. Ethernet is much more flexible, it may be used in a similar master-slave fashion, and it also allows module to module transfers and broadcasting. To provide reliable collision detection ethernet cannot transfer packets smaller than sixty-four bytes in length. However, the inability to send single word packets is not significant since the data transfer scheme was adjusted to make use of the extra data bytes available. The data transfer scheme was further enhanced to make use of the advantages of ethernet by adding data-driven communication to the master-slave communication concept that was used previously by the BPI data bus. A data-driven scheme allows the module that has new data to send it immediately rather than waiting for the master computer to fetch data based upon a cyclic poll of the modules.

Since the host computer is a UNIX based machine with TCP/IP over ethernet networking protocols built in, the data encapsulation protocol was chosen from the TCP/IP suite. TCP/IP has Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP is connection-oriented and reliable, whereas UDP is connectionless and has no built in error correction. The wind tunnel data acquisition system modules are located well within the distances allowed on a single segment of ethernet. In this dedicated small network it was anticipated that the number of lost, out of sequence or erroneous packets would be very low. Since a single segment network negated the reliability advantage of TCP, and as UDP can broadcast packets, which connection-oriented TCP cannot do, and because UDP appeared easier to implement into RTK, UDP was chosen to be the communication protocol.

UDP protocol encapsulates the data by placing an eight-byte header in front of the data. The UDP header specifies the source and destination ports, a total length and a check-sum. When the host initiates communication with a module, the master-slave port number, 40312, is used as the destination port. When modules send unsolicited data to the host or other modules, the data-driven port number, 40311, is used as the destination port. The information carried in the UDP encapsulated packets can be divided into two sections. First, there is a header section, which is followed by a data section. The structure of the header section is shown below.

[3] 10 Megahertz Base-band signalling over Twisted pair wiring.

```
struct LSWT_UDP_HDR {
    unsigned short int  commandOrCommandBeingRepliedTo; /* ie 0x0062 is module ID */
    unsigned short int  sequenceNumber;                 /* allows some error checks */
    unsigned short int  commandResponse;                /* Zero if no error */
    unsigned short int  actualDataTypeAndLength;        /* Data type and Length */
    unsigned long  int  timeSinceMidnightInMilliseconds; /* Zero if not available */
    unsigned short int  shortDummy_1;                   /* For future - undefined */
    unsigned short int  shortDummy_2;                   /* For future - undefined */
    unsigned long  int  longDummy_1;                    /* For future - undefined */
    unsigned long  int  longDummy_2;                    /* For future - undefined */
}
```

Each packet sent from the host to the modules, and vice-versa, using the master-slave port carries the header structure above, plus an eighty-byte data section, a total length of 104 bytes. The information in the header indicates the type of data and the actual length of the data. Data-driven mode packets use an extended data section appended to the same header structure. This data section contains all the data held by the module plus an identification string. If all the string space is used then the total length is 264 bytes.

Modules in data-driven mode can 'narrowcast' (unicast) the data instead of broadcasting it to reduce module overheads from processing irrelevant network traffic. Broadcast mode is useful for modules that have information required by other modules as well as the host.

The user interface software programs developed on the host computer run as single processes independent of one another. Each program must be able to access the most current data that is available from the modules. Additionally, only one program (process) can bind to a specific UDP port at a time. To overcome these constraints, shared memory was chosen as the interprocess communication method, as it would provide data sharing. Shared memory allows two or more processes to have a region of memory that each process may reference. Once each process sets up the shared memory, the processes can access the data in shared memory without involving the kernel at all. Synchronisation must be provided when accessing the shared memory and this is handled using semaphores. (Stevens, 1999)

The program *testBcast* listens to the data-driven UDP port and updates the shared memory as modules send their data over the network. The *rdData* library routines provide a calling program with the latest data sent by a particular module. These routines read the data from the shared memory that is continually updated by the *testBcast* program. A semaphore is used to control access to the shared memory region so that the data is not allowed to change part way through a read access.

The shared memory is divided into a control area and a data area. The semaphore is used by *testBcast* and *rdData* routines to force the update, of any information in the control area, to be strictly one process at a time. Information in the control area maps

each module in data-driven mode to a section of the data area. The location of the section within the data area changes as each packet arrives. When a packet is received it is directly written to an unused section of the data area. This new packet of information is then made available to any new read requests when *testBcast* next obtains the semaphore and updates the control area for the module concerned. Packet information made obsolete by the new arrival is not reclaimed for reuse until all read requests on it are finished. A read request through an *rdData* routine increments a counter in the control area associated with a section of the data area when it starts and decrements the same counter when it finishes. This allows *testBcast* to find and reuse an obsolete section by reading the value in its counter.

8. Data Transfer Rate Comparison and Reliability

A comparison of the data transfer rates of the ethernet and the previous BPI data bus was performed (Edwards, 1999) and the results are summarised in Table 1. The BPI rates measured were based on the time taken to fetch a single address (vector), whereas the ethernet rate is fundamentally a packet update rate measured by the host computer. The data bus rate for the ethernet was determined by knowing the number of addresses fetched in a single ethernet packet. The results shown in the table illustrate that the ethernet bus data transfer rates far exceed those achieved by the previous BPI data bus. One of the two main upgrade objectives, to improve the data transfer rates over the new bus, was therefore achieved.

Module	Time per address (s)		Addresses per Second	
	Ethernet	BPI (old system)	Ethernet	BPI (old system)
Freestream	0.0014	0.0574	720	17
Sting Column		0.0555		18
Auxiliary	0.0016	0.0700	640	14
Strain Gauge #1	0.0003	0.0642	3280	16
Strain Gauge #2	0.0004		2560	
Model Attitude	0.0005	0.0700	2000	14
Inclinometer	0.0002	0.1200	4160	8
Actuator	0.0001	0.0526	7680	19

Table 1. Comparison of data transfer rates for the ethernet and BPI data bus

The *testBcast* software monitors the packet traffic on the ethernet and maintains a count of the number of lost packets. Over the extended use of the ethernet data bus it has been found that the loss of UDP packets is insignificant, being of the order of one lost packet in a million packets transferred.

The ethernet network has also proved itself to be reliable. The previous BPI based system was prone to dropouts and data loss due to the high electrical interference in

the wind tunnel operating environment. However, the new ethernet network has not been disturbed by the electrical interference generated by other systems in the wind tunnel. Since its integration with the data acquisition system, the ethernet data acquisition network has not crashed at all. The second main objective, to improve the reliability of the data bus, was therefore also achieved.

9. VXI-based Instrumentation

The development of the data acquisition system is a longer-term project with the objectives of maintaining a reliable, state-of-the-art system, which provides high quality data to the wind tunnel customer. With this in mind, the data acquisition system project has commenced the transition of the now outdated VME-based instrumentation modules to a new off-the-shelf system. VXIbus was chosen as the development platform for the possible replacement of some of the instrumentation modules.

VXIbus was initially introduced in 1987 as a new standard instrument architecture. It is defined around the VMEbus architecture which is well known for its high speed data rates of up to 40 MB/s, and its excellent computer backplane. VXIbus took the VMEbus standard and developed it further to incorporate GPIB; to define the instrumentation environment including power and cooling required, with strict limits on radiated interference between modules; to require a Slot 0 backplane management card; and to define a Resource Manager to configure modules when power is turned on or reset. VXIbus is now known for its compact size, high throughput and flexibility. (Hewlett-Packard, 1998)

Additionally, VXIbus was chosen by the prime Contractor as the backbone of the data acquisition system in the new AMRL Transonic Wind Tunnel. To maintain commonality, and transferability between the two major wind tunnels, it was decided to pursue the VXIbus development path.

The strain gauge balance amplifier system, as shown in Figure 6, was chosen as the first of the VME-based modules to be investigated as it predominantly transfers data continuously to the host computer, and it does not incorporate any control functions. Six Vishay model 2310Y signal conditioning amplifiers were acquired to provide conditioning and amplification of the low-level signals from the six strain gauge bridges on a balance. The excitation voltage and bridge output voltage for each component were connected into a Hewlett Packard E1415 Closed Loop Algorithmic Controller A/D card housed in the VXI main chassis.

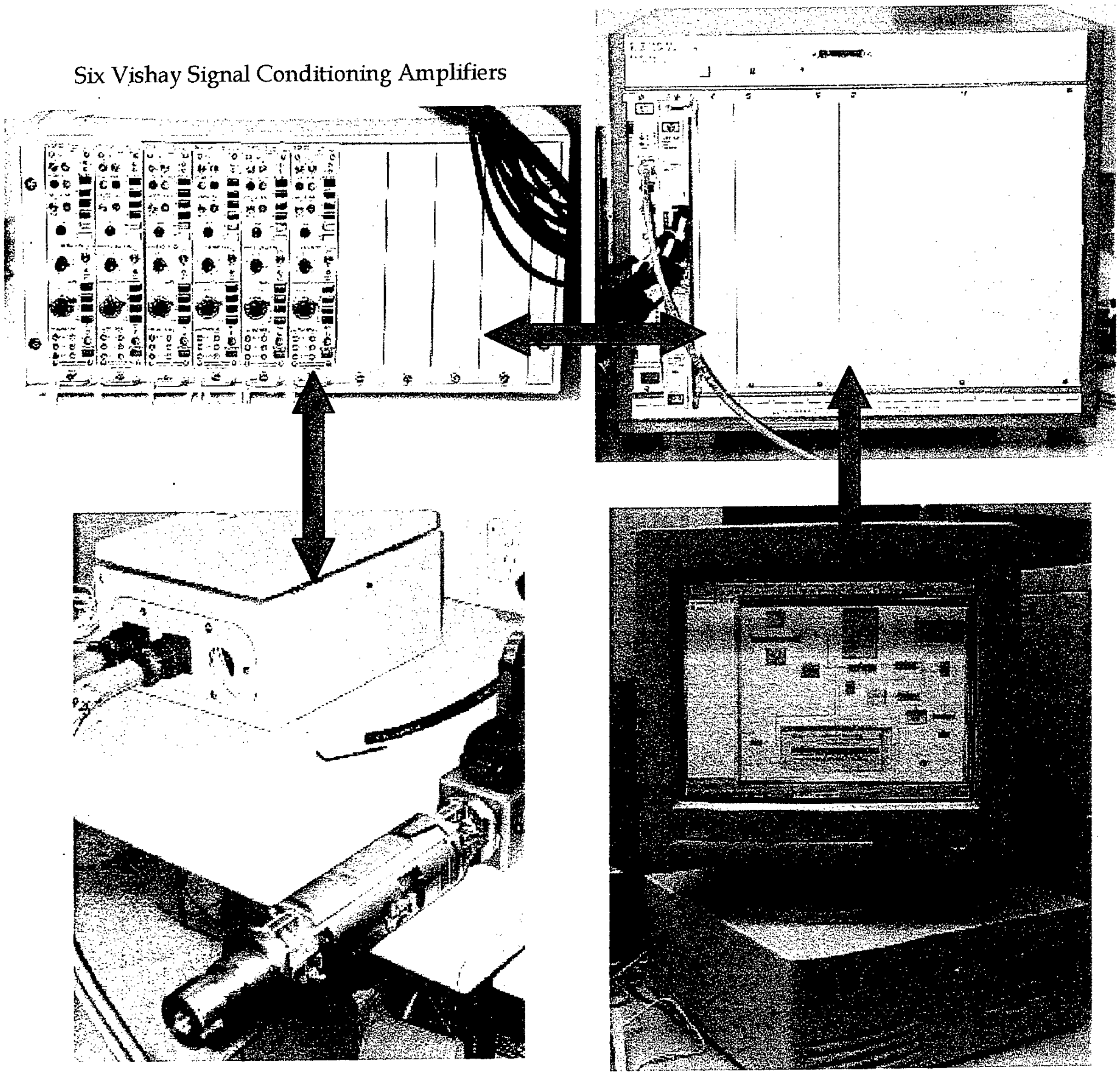


Figure 6 VXI-based strain gauge balance instrumentation setup

An important decision regarding the VXIbus system was the choice of the Slot 0 controller. The options to choose from are GPIB, Firewire (IEEE-1394), MXI-2, or embedded PC controller. The most compact, highest throughput, and also the most expensive is the embedded PC controller. However, no upgrade path (other than purchasing a new PC controller) is available with this option. This is a major drawback as PC technology is advancing very rapidly. Firewire was chosen as the preferred

controller based on cost and throughput requirements. So far, data update rates indicate that the Firewire is providing the throughput required. The Firewire controller connects to a PCI card located in a 500 MHz Pentium III PC, which acquires the data and sends it to the host computer via the ethernet.

As described in previous sections of this report, the wind tunnel data acquisition system is based around a dedicated ethernet network. As one of the prime objectives of the project was to provide ease of maintainability, HP VEE software was chosen as the development platform for acquisition of data from the A/D card, and transmission of the data to the host computer. HP VEE provides an object based software development environment, with the necessary drivers, and objects, to communicate directly with the instrumentation cards and TCP/IP. This made the software development and integration process extremely smooth. A daemon running on the Digital Alpha Server 400 host computer was written to receive the TCP/IP data packets from the VXI host PC and place them in a shared memory area for processes such as *monfrc* or *comfrc* to read as required.

The VXI-based system is currently providing data to the host computer at an update rate of 40 channels at 50 Hz. This rate can easily be increased if required, however, 50 Hz has been chosen to be consistent with the other modules in the system.

10. Conclusion

This report describes the upgrade of the data acquisition system in the Low Speed Wind Tunnel at the Aeronautical and Maritime Research Laboratory. The new system is based on a Digital AlphaServer 400 running Digital UNIX as the host computer, which provided a C, X/Motif, and OpenGL software development environment. Using this mixture of development tools the system has been designed with platform independence and ease of maintenance as key design features.

The bi-directional parallel interface data bus, designed in the late 1980s and used in the previous data acquisition system, did not achieve its stated speed or reliability design objectives and it has now been replaced with the widely accepted TCP/IP over ethernet protocols. The ethernet based data bus has proved to be both reliable and significantly faster than the previous data bus.

A VXI-based data acquisition system is currently being evaluated as a replacement for some of the slave modules used in the wind tunnel data acquisition system. As the VME-based modules are becoming more difficult to maintain, and as data acquisition technology continues to progress, the next version of the wind tunnel data acquisition system will most likely see a mixture of PCs and VXI, using ethernet and firewire as the communication protocols.

11. Acknowledgements

This project has been running since July 1996 and has achieved many major milestones due to the effort of the Low Speed Wind Tunnel Data Acquisition team. This team is made up of: Fred Bird, Craig Edwards, Owen Holland, Steve Kent, Stephen Lam, Yoel Link, Neil Matheson, Robert Toffoletto, Mike Spataro and Reg Worl. Without the combined effort of all the team members this project would not have succeeded in its objectives.

12. References

Bird, F., 1999, *A Freestream Parameter Instrumentation Module for the AMRL Low Speed Wind Tunnel,* to be published, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Edwards, C.D., Link, Y.Y., 1999, *COMFRC: Software for the Acquisition of Data in the Low Speed Wind Tunnel Data Acquisition System,* to be published, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Edwards, C.D., 1999, *MicroVAX/Bus Communication Testing Software for the Low Speed Wind Tunnel Data Acquisition System,* DSTO-TN-0189, AR-010-843, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Edwards, C.D., 1999, *MONFRC: Software for the Graphical Display of Parameters and Data in the Low Speed Wind Tunnel Data Acquisition System,* DSTO-TN-0218, AR-011-056, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Edwards, C.D., 1999, *WTSETUP: Software for Creating and Editing Configuration Files in the Low Speed Wind Tunnel Data Acquisition System,* DSTO-TN-0217, AR-011-055, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Fairlie, B.D., Lam, S.S.W., 1993, *A Software Interface for the ARL Wind Tunnel Data Acquisition System,* ARL-TR-14, AR-007-135, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Harvey, J., 1989, *A Data Acquisition Parallel Bus for Wind Tunnels at ARL,* ARL-FLIGHT-MECH-TM-412, AR-005-629, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Harvey, J.F., Sutton, C.W., Bird, F.J., 1989, *A Microprocessor Controlled Strain Gauge Calibration Module,* ARL-FLIGHT-MECH-TM-422, AR-005-579, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Hewlett-Packard, 1998, *1998 Test System and VXI Products Catalog*, Hewlett-Packard Company, USA.

Kent, S., 1993, *A Wind Tunnel Model Control Surface Actuator Interface*, ARL-TN-13, AR-006-658, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Kent, S., 1998, *A Computer Control Interface to Operate Turntables in the Test Section of a Wind Tunnel*, DSTO-TR-0622, AR-010-528, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Lam, S.S.W., Link, Y.Y., 1993, *Development and Operation of the F/A-18 Model Control Surface Actuators*, ARL-TN-29, AR-008-345, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Matheson, N., Harvey, J., Lam, S., 1991, *A Parallel Bus Wind Tunnel Data Acquisition System*, Presented at the 75th Semi-annual STA meeting, Austin, Texas, USA.

Motorola, 1985, *The VMEbus Specification Manual, HB212 Rev. C.1*, Second Printing, USA.

Spataro, M., Kent, S., 1998, *A Serial Communication Interface for Data Acquisition Instrumentation in a Wind Tunnel*, DSTO-TR-0740, AR-010-669, DSTO Aeronautical and Maritime Research Laboratory, Australia.

Stevens, W.R., 1999, *UNIX Network Programming Interprocess Communication*, Prentice-Hall, USA.

Toffoletto, R., 1998, *A New Actuator Control Program for the Low Speed Wind Tunnel*, DSTO-TN-0188, AR-010-841, DSTO Aeronautical and Maritime Research Laboratory, Australia.

DISTRIBUTION LIST

A New Data Acquisition System for the AMRL Low Speed Wind Tunnel

Owen Holland, Stephen Lam and Yoel Link

AUSTRALIA

DEFENCE ORGANISATION

S&T Program

Chief Defence Scientist
FAS Science Policy } shared copy
AS Science Corporate Management
Director General Science Policy Development
Counsellor Defence Science, London (Doc Data Sheet only)
Counsellor Defence Science, Washington (Doc Data Sheet only)
Scientific Adviser Policy and Command
Navy Scientific Adviser (Doc Data Sheet and distribution list only)
Scientific Adviser - Army (Doc Data Sheet and distribution list only)
Air Force Scientific Adviser
Director Trials

Aeronautical and Maritime Research Laboratory

Director

Chief of Air Operations Division: C. Guy
Research Leader: N. Pollock
Head: N. Matheson (3 copies)
O. Holland (3 copies)
S. Lam (3 copies)
Y. Link (3 copies)

G. Ainger
F. Bird
J. Clayton
L. Erm
M. Glaister
J. Harvey
P. Malone
M. Spataro
R. Worl
I. Amott
D. Carnell
C. Edwards
B. Fairlie
A. Gonzalez
S. Kent
H. Quick
R. Toffoletto

DSTO Library and Archives

Library Fishermans Bend
Library Maribyrnong
Library Salisbury (2 copies)
Australian Archives
Library, MOD, Pyrmont (Doc Data sheet only)
*US Defense Technical Information Center, 2 copies
*UK Defence Research Information Centre, 2 copies

*Canada Defence Scientific Information Service, 1 copy
*NZ Defence Information Centre, 1 copy
National Library of Australia, 1 copy

Capability Development Division
Director General Maritime Development (Doc Data Sheet only)
Director General Land Development (Doc Data Sheet only)
Director General C3I Development (Doc Data Sheet only)
Director General Aerospace Development (Doc Data Sheet only)

Army
ABCA Standardisation Officer, Puckapunyal, (4 copies)
SO (Science), DJFHQ(L), MILPO Enoggera, Queensland 4051 (Doc Data Sheet only)
NAPOC QWG Engineer NBCD c/- DENGRS-A, HQ Engineer Centre Liverpool Military Area, NSW 2174 (Doc Data Sheet only)

Intelligence Program
DGSTA Defence Intelligence Organisation
Defence Intelligence Organisation-Information Centre

Corporate Support Program
OIC TRS, Defence Regional Library, Canberra

University and Colleges

Australian Defence Force Academy	• Library • Head of Aerospace and Mechanical Engineering
Adelaide	• Barr Smith Library
Flinders	• Library
LaTrobe	• Library
Melbourne	• Engineering Library
Monash	• Hargrave Library
Newcastle	• Library
New England	• Library
Sydney	• Engineering Library
NSW	• Physical Sciences Library
Queensland	• Library
Tasmania	• Engineering Library
Western Australia	• Library
RMIT	• Library
University College of the Northern Territory	• Library

OTHER ORGANISATIONS

NASA (Canberra)
AGPS

OUTSIDE AUSTRALIA

ABSTRACTING AND INFORMATION ORGANISATIONS

Library, Chemical Abstracts Reference Service
Engineering Societies Library, US
Materials Information, Cambridge Scientific Abstracts, US
Documents Librarian, The Center for Research Libraries, US

INFORMATION EXCHANGE AGREEMENT PARTNERS

Acquisitions Unit, Science Reference and Information Service, UK
Library - Exchange Desk, National Institute of Standards and Technology, US

SPARES (5 copies)

Total number of copies: **86**

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA					1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)
2. TITLE A New Data Acquisition System for the AMRL Low Speed Wind Tunnel			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)		
4. AUTHOR(S) Owen Holland, Stephen Lam and Yoel Link			5. CORPORATE AUTHOR Aeronautical and Maritime Research Laboratory PO Box 4331 Melbourne Vic 3001 Australia		
6a. DSTO NUMBER DSTO-TR-0896	6b. AR NUMBER AR-011-129		6c. TYPE OF REPORT Technical Report	7. DOCUMENT DATE November 1999	
8. FILE NUMBER M1/9/720	9. TASK NUMBER RDI 98/179	10. TASK SPONSOR DST	11. NO. OF PAGES 18	12. NO. OF REFERENCES 17	
13. DOWNGRADING/DELIMITING INSTRUCTIONS			14. RELEASE AUTHORITY Chief, Air Operations Division		

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

low speed wind tunnels, data communicating systems, VMEbus (computer bus), ethernet (local area network system), VXIbus (computer bus), wind tunnels, data acquisition, computer systems

19. ABSTRACT
The data acquisition system in the Low Speed Wind Tunnel at the Aeronautical and Maritime Research Laboratory was recently upgraded. The MicroVAX II host computer was replaced by a Digital AlphaServer 400 running Digital UNIX, and the Bi-directional Parallel Interface data bus was replaced by ethernet and fast serial communication. The upgrade provides a system which is easier to use; includes a graphical user interface; provides a communication bus based on standard communication protocols, which achieves higher data transfer rates; is far more reliable and easy to maintain; and the system is more flexible than previous versions.

Page classification: UNCLASSIFIED